

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN INGENIERÍA INFORMÁTICA

Trabajo Fin de Grado
Mantenimiento evolutivo de un juego
educativo para ordenador

Autor: Jesús Labarga de Navarro
Director/es: M. Carmen Pagés Arévalo

TRIBUNAL:

Presidente:

Vocal 1º:

Vocal 2º:

FECHA:

Dedicado a mi familia y amigos
y agradecimientos a la profesora Carmen Pagés Arévalo.

Índice

Resumen.....	9
Summary.....	9
Palabras clave.....	9
Resumen extendido	11
Introducción.....	13
Base Teórica	13
Mantenimiento software	13
ECDL.....	15
E-learning y G-learning.....	16
Arquitectura	17
Diagrama de clases	17
Descripción de las clases principales de la aplicación	18
Gestor de preguntas	31
Modelo de datos	32
Fichero XML	34
Herramientas y tecnologías utilizadas	35
Entorno de desarrollo	35
Lenguaje de programación.....	35
Base de datos	35
Máquina virtual	36
Control de versiones y repositorio	36
Mantenimiento correctivo	37
El error del menú y los módulos.....	37
El error de las imágenes y la lupa.....	39
El error de las preguntas repetidas.....	42
Añadir sonidos al juego	43
Mantenimiento Evolutivo	45
Colaboración en línea – Módulo 2	46
Objetivos del módulo.....	46
Ejemplo de preguntas sobre este módulo	47
Seguridad - Módulo 8.....	49
Objetivos del módulo.....	49
Ejemplo de preguntas sobre este módulo	50
Actualización del documento de ayuda	52

Pruebas.....	53
Conclusiones.....	55
Bibliografía.....	57

Índice de diagramas tablas y figuras

1 Ciclo de vida del software	13
2 Distribución del ciclo de vida por coste	14
3 Distribución del tiempo en el mantenimiento.....	15
4 Diagrama de clases.....	17
5 Gestor de preguntas.....	31
6 Modelo de datos	33
7 Fichero XML.....	34
8 ECDL PI@y sin nombre de módulos	38
9 ECDL PI@y con nombre de módulos.....	39
10 ECDL PI@y con icono de lupa.....	41
11 ECDL PI@y sin icono de lupa	41
12 Tabla cambios de módulos.....	45
13 Pantalla inserción preguntas	46
14 Tabla pruebas.....	54

Resumen

Mantenimiento evolutivo y correctivo del juego ECDL PI@y. ECDL PI@y es un juego educativo de preguntas y repuestas de tipo tablero por turnos, la finalidad del juego es facilitar la obtención de la Acreditación Europea de Manejo de Ordenador o ECDL

La parte de mantenimiento evolutivo consiste en la introducción y creación de nuevas preguntas según los nuevos módulos ECDL así como la actualización de los módulos anteriores.

El mantenimiento correctivo soluciona los errores de programación encontrados en la versión anterior del juego educativo así como la creación de nuevas funcionalidades.

Summary

Evolutionary and corrective maintenance of the game ECDL PI@y. ECDL PI@y is an educational game of question and answer type board in turn, the object of the game is easier to obtain the European Computer Driving Accreditation or ECDL

The part of evolutionary maintenance is the introduction and creation of new questions as the new ECDL modules and updating the previous modules.

Corrective maintenance fixes the bugs found in the previous version of the educational game and the creation of new functions.

Palabras clave

Mantenimiento, Juego Educativo, eLearning, ECDL

Resumen extendido

El trabajo consiste en el mantenimiento evolutivo y correctivo de la aplicación desarrollada en un trabajo anterior llamada ECDL PI@y. ECDL PI@y es un juego educativo de preguntas y repuestas de tipo tablero por turnos. Los temas de las preguntas están basadas en el temario Syllabus para ayudar a aprender y conseguir la acreditación internacional ECDL. ECDL otorga el reconocimiento de poseer una formación básica y completa en informática a nivel de usuario.

En la parte de mantenimiento correctivo nos encontramos con tres errores principales tras realizar las pruebas pertinentes que había que solucionar.

El primero de ellos consistía en la corrección del menú de módulos, en la versión anterior en el menú de módulos aparecían los módulos sin su nombre lo que suponía no saber qué tema tendrían las preguntas de la partida.

El segundo error era que en las preguntas con imagen se tenía que pulsar un icono de una lupa para que apareciese la imagen de la pregunta, algo incómodo, además en la siguiente pregunta respecto a la de la imagen continuaba apareciendo el icono de la lupa y si se le pulsaba aparecía la imagen de la pregunta anterior. Se optó por solucionar ese error quitando la lupa y corrigiendo el problema de la imagen en una pregunta posterior.

El ultimo error y más importante es el de la repetición de preguntas, lo que ocurría era que se repetían las preguntas respondidas en un breve periodo de tiempo, algunas incluso seguidas. Se descubrió que era por el sistema que tenía el juego de repartir las preguntas el cual no estaba bien refinado, esto se arregló con unas listas de preguntas.

Además de estos errores, al juego se le implemento varios sonidos para hacer la partida más amena y entretenida.

Respecto a la parte del mantenimiento evolutivo se han revisado los módulos anteriores, como el temario cambiaba en algunos módulos se han creado nuevas preguntas, además también se fusionaron con el nuevo temario dos módulos en uno. También se han creado dos módulos nuevos con sus 120 preguntas cada uno repartidas por nivel de dificultad y teniendo un 30% de ellas imágenes.

También se ha actualizado el menú de módulos incorporando los dos módulos nuevos.

Por último se realizaron las pruebas pertinentes para comprobar que los cambios realizados, además de que funcionasen bien, no influyeron negativamente a las otras partes de la aplicación anterior.

Introducción

ECDL PI@y es un juego de la oca en el que los jugadores deben responder a una pregunta cuando les toca el turno y si aciertan tiran el dado y avanzan y si no aciertan se quedan en la misma casilla.

El objetivo principal del trabajo hacer el mantenimiento evolutivo del juego pero también se realizara un mantenimiento correctivo.

La parte de mantenimiento evolutivo consiste en la introducción y creación de nuevas preguntas según los nuevos módulos ECDL.

El mantenimiento correctivo solucionará los errores de programación encontrados en la versión actual del juego educativo así como la creación de nuevas funcionalidades.

Base Teórica

Mantenimiento software

El mantenimiento de software es la modificación de un producto de software después de la entrega, para corregir errores, mejorar el rendimiento, u otros atributos.

El mantenimiento de software es una de las fases en el ciclo de vida de desarrollo de software.

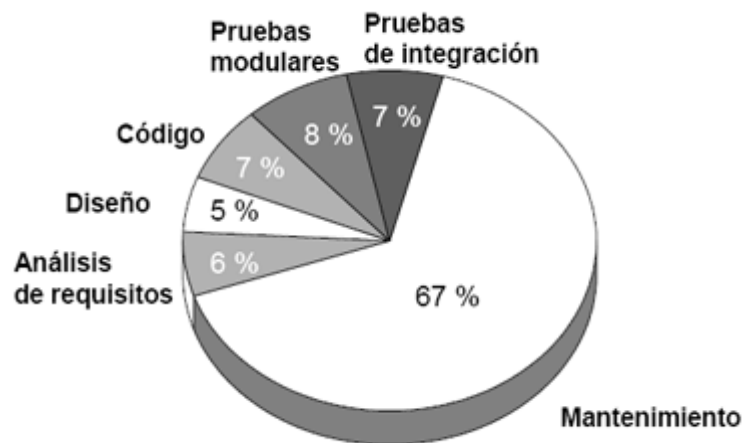
El ciclo de la vida de desarrollo de software lo podemos ver en la siguiente imagen:



1 Ciclo de vida del software

El mantenimiento del software es una tarea infravalorada e importante que la mayor parte del coste del ciclo de vida del software.

En el siguiente grafico podemos ver la distribución del coste del ciclo de vida en sus etapas:



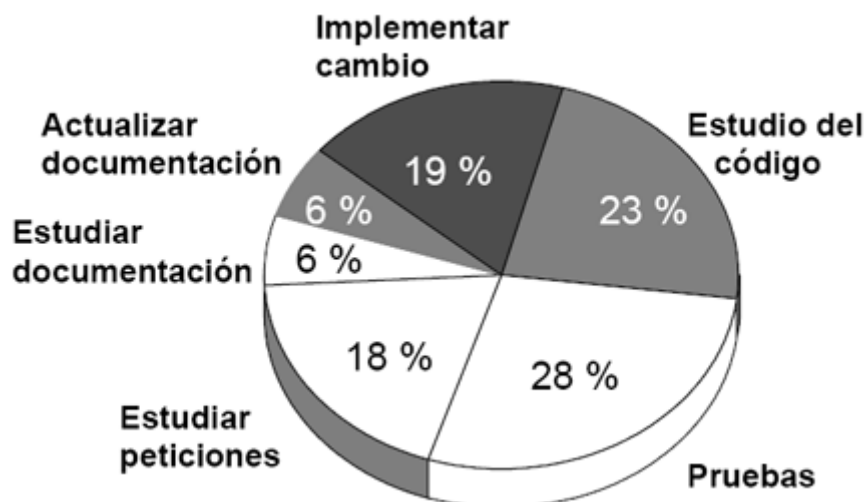
2 Distribución del ciclo de vida por coste

Este mayor coste es debido a múltiples factores, entre los que podemos encontrar:

- Inexistencia de métodos, técnicas y herramientas
- Continúas modificaciones.
- Documentación mala o inexistente.
- Infravaloración del mantenimiento.
- Presión de tiempo.
- Poca participación del usuario.

Las actividades más comunes en mantenimiento son las siguientes:

- Corrección de defectos en el software.
- Creación de nuevas funcionalidades en el software por nuevos requisitos de usuario.
- Mejora de la funcionalidad y del rendimiento



3 Distribución del tiempo en el mantenimiento

Como se puede observar en el gráfico anterior, el mantenimiento en sí mismo es como el ciclo de vida global ya que se divide en etapas parecidas.

ECDL

ECDL es la acreditación internacional europea que otorga el reconocimiento de poseer una formación básica y completa en informática a nivel de usuario. Gestionada por la Fundación ECDL, la acreditación está implantada en toda Europa y, bajo las siglas ICDL, en el resto del mundo.

El destinatario de la ECDL es toda la ciudadanía en general que, con independencia de su nivel académico y de su profesión, desee o necesite acreditar sus conocimientos y habilidades sobre las Tecnologías de la Información.

Surgida como una iniciativa del Consejo Europeo de Asociaciones Profesionales de Tecnologías de la Información (CEPIS) para promover y aumentar la competencia de los europeos en el uso de las Tecnologías de la Información, la acreditación ECDL ha sido recomendada por la Comisión Europea y goza de reconocimiento oficial en varios países por parte de algunas de sus administraciones. En España, la implantación de la ECDL está supervisada y canalizada a través de la Asociación de Técnicos de Informática.

La acreditación se obtiene tras superar un test que puede realizarse en cualquiera de los Centros de Pruebas Homologados repartidos por todo el estado. Los conocimientos y habilidades requeridos a los candidatos están claramente estructurados y detallados en el temario Syllabus.

E-learning y G-learning

El e-learning consiste en la educación a través de Internet y las nuevas tecnologías. Este tipo de enseñanza online permite la interacción del usuario con el material mediante la utilización de diversas herramientas informáticas.

Los beneficios del e-learning son:

- Reducción de costos: permite reducir y hasta eliminar gastos de traslado, alojamiento, material didáctico, etc.
- Rapidez y agilidad: Las comunicaciones a través de sistemas en la red confiere rapidez y agilidad a las comunicaciones.
- Acceso just-in-time: los usuarios pueden acceder al contenido desde cualquier conexión a Internet, cuando les surge la necesidad.
- Flexibilidad de la agenda: no se requiere que un grupo de personas coincidan en tiempo y espacio.

A pesar de que el e-learning aporta innumerables ventajas, esta modalidad de formación presenta cierta ineficacia en lo que al desarrollo de habilidades se refiere.

A través del e-learning se pueden ofrecer contenidos muy completos, interactivos, dinámicos, actualizados, así como sistemas de evaluación de contenidos en tiempo real.

Pero todas estas ventajas, que son muy útiles de cara a facilitar la transmisión y asimilación de conceptos, no contribuyen al desarrollo de una habilidad.

Para adquirir éstas es necesario que el alumnado pueda probar, practicar, experimentar, ensayar, de forma casi ilimitada, y, al mismo tiempo, recibir un *feedback* prácticamente inmediato, pudiendo observar las consecuencias de sus decisiones. Es aquí donde entra en juego, nunca mejor dicho, el g-learning.

EL g-learning no deja de ser formación e-learning pero que incorpora un simulador que permite poner en práctica los conocimientos adquiridos.

Varios estudios sostienen que el g-learning mejora el pensamiento estratégico, la atención, la velocidad para la toma de decisiones y la capacidad para la multitarea

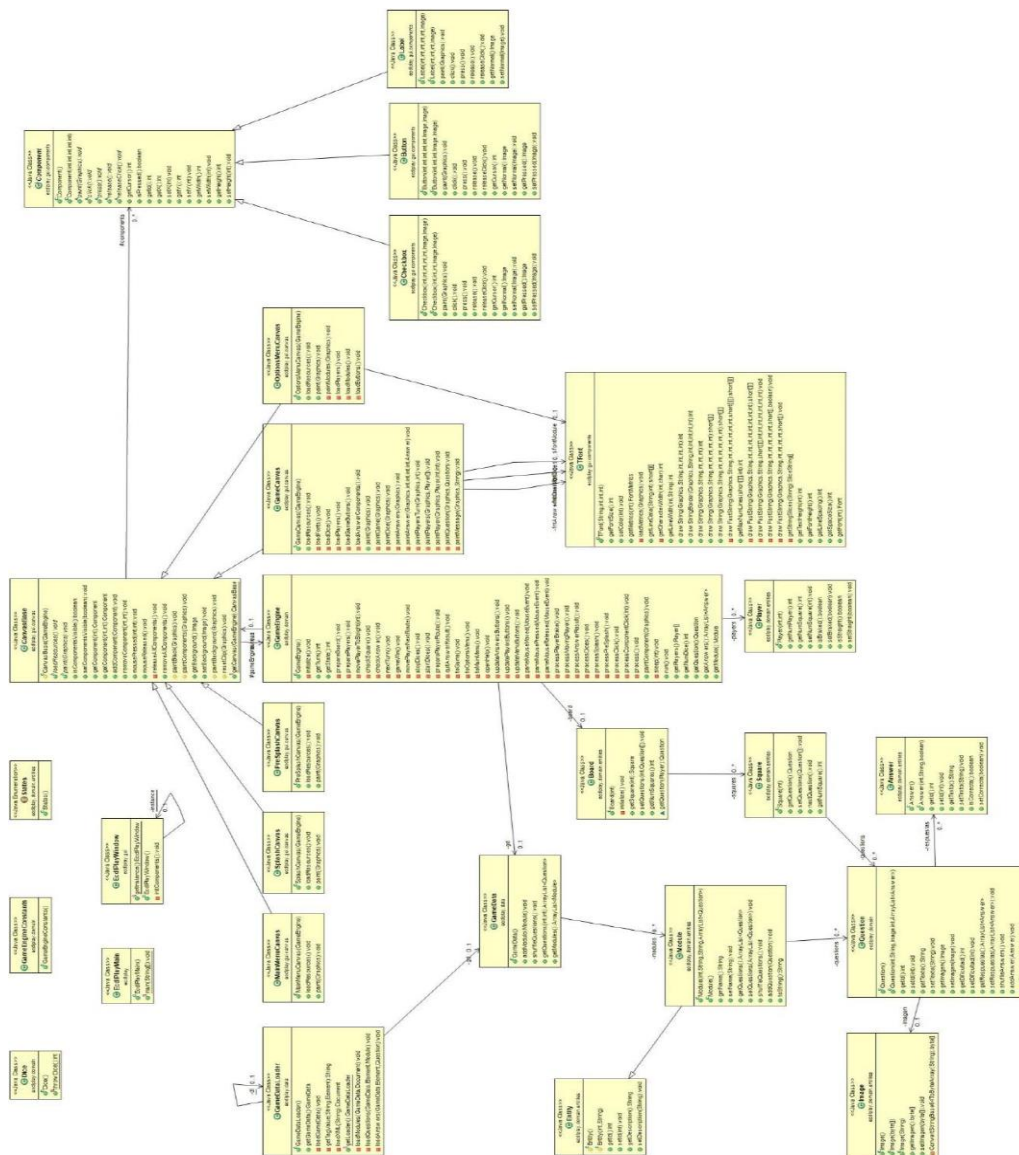
Arquitectura

El juego ECDL pl@y está basado en dos aplicaciones, una que es el juego propiamente dicho y otra el gestor de preguntas. El gestor de preguntas es un programa para manejar la base de datos, es un programa web que genera un fichero XML con las preguntas el cual es cargado en la aplicación del juego.

El mantenimiento correctivo se aplicará sobre la aplicación del juego y el evolutivo sobre la base de datos.

Diagrama de clases

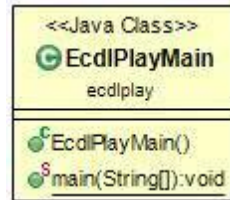
Diagrama de clases de la aplicación del juego:



4 Diagrama de clases

Descripción de las clases principales de la aplicación

EcdIPlayMain



Clase main de la aplicación. Encargada de ser el punto de entrada de ejecución y cargar la ventana de la aplicación.

Métodos	
static void	main (String[] args) Ejecuta el programa.

GameEngine



Clase principal de la aplicación. Encargada de la lógica de la aplicación como la máquina de estados del menú y juego, lógica del juego y comunicación con las clases canvas para el pintado de los componentes.

Atributos	
private Board	board Instancia del tablero, que se inicializa cada vez que comienza un nuevo juego
private CanvasBase	canvas Instancia de la clase canvas que se utiliza para pintar la pantalla
private GameData	gd Instancia de la clase GameData de la que se obtienen los datos de la aplicación
private long	lastProcessTime Indica la última “posición del mundo” procesada por el hilo de repintado
private int	maxPlayerRoute Indica en qué posición de PlayerRoute estamos en cada momento
private int	module Identificador del módulo sobre el que se realizarán las preguntas del juego.
private int	numAnswer Respuesta seleccionada por el jugador
private int	numDice Identificador de la respuesta que ha elegido el usuario
private int	numPlayers Número de jugadores que participarán en el juego.
private Point[]	playerRoute Valor devuelto por el dado en una tirada
int	playerRouteOffsetX Posición X del punto actual de la ruta
int	playerRouteOffsetY Posición Y del punto actual de la ruta
private int[]	playerRouteSquares Array de puntos de pantalla para generar la ruta que tiene que seguir la animación de avanzar casillas de un jugador
private Player[]	players Array de los jugadores jugando la partida.
private int	posPlayerRoute Indica en qué posición de PlayerRoute estamos en cada momento
private boolean	running Indica si el hilo de repintado tiene que parar (running = false) o no
private int	state Indica el estado actual de la máquina de estados

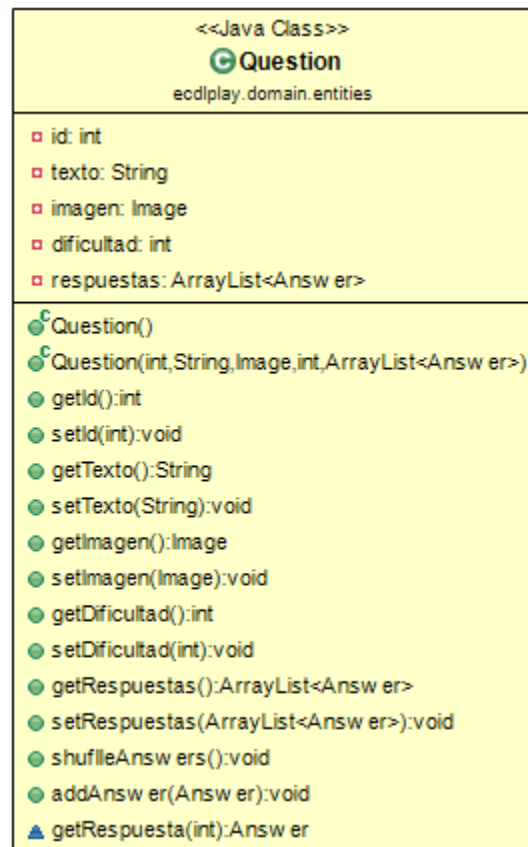
private Thread	thread Instancia al objeto Thread que se encarga de repintar la pantalla cada poco tiempo.
private long	timeDices Tiempo en el que los dados tienen que dejar de estar moviéndose
private long	timeFinish Variable que se suele utilizar para marcar un tiempo en que la aplicación tiene que hacer algo, como por ejemplo saltar el Splash después de unos segundos.
private int	turn Indica el turno actual, quién es el jugador jugando en cada momento

Métodos	
private void	checkAnswer() Comprueba si la respuesta indicada por el jugador es correcta
private void	checkSquare() Comprueba la casilla en la que ha caído el jugador actual, para comprobar si es una casilla de frenada, de escalera, de fin o una casilla normal
private void	gameWin() Cambia el estado indicando que un jugador ha llegado a la casilla final
ArrayList<Answer>	getAnswers() Obtiene las respuestas de la pregunta de la casilla actual
Module	getModule() Obtiene la información del módulo sobre la que se van a hacer las preguntas
int	getNumDice() Obtiene el valor devuelto por el dado
Player[]	getPlayers() Obtiene los jugadores actuales
Question	getQuestion() Obtiene la pregunta de la casilla actual
int	getState() Devuelve el estado actual de la máquina de estados
int	getTurn() Devuelve el turno del jugador que está jugando en un momento determinado
private void	initialize() Se encarga de inicializar el motor gráfico.
private void	movePlayerNextRoute()

	Mueve la pieza del jugador a la siguiente casilla en la ruta
private void	movePlayerToStraight(int numSquare) Avanza la pieza del jugador hasta la casilla indicada por el parámetro
private void	nextTurn() Cambia el turno al siguiente jugador, mostrando la pregunta y las respuestas
private void	openHelp() Abre el PDF asociado a la ayuda del juego
void	paintComponent(Graphics g) Se llamada cada vez que hay que Java necesita repintar.
private void	panelMouseClicked(event.MouseEvent evt) Se lanza cada vez que el usuario mueve el ratón por la ventana.
private void	panelMousePressed(event.MouseEvent evt) Se lanza cada vez que el usuario presiona un botón del ratón.
private void	panelMouseReleased(event.MouseEvent evt) Se lanza cada vez que el usuario libera un botón del ratón.
private void	prepareBoard() Inicializa el tablero, asignando las preguntas a cada una de las casillas
private void	preparePlayerRoute() Inicializa la ruta que seguirá la ficha del jugador según el número de casillas que tenga que avanzar
private void	preparePlayers() Inicializa el array de jugadores así como el array de rutas y de casillas
private void	process() Proceso principal que se encarga de llamar a los sub procesos de gestión dependiendo del estado actual de la máquina de estados.
private void	processAnswerResult() Hace de temporizador para mostrar el mensaje de Respuesta Correcta/Incorrecta durante unos segundos
private void	processClick() Procesa el click del ratón teniendo en cuenta en el estado en que se encuentra la máquina de estados.
private void	processComponentClick(int id) Procesa el click de ratón en algún componente gráfico de pantalla como un botón un Checkbox.
private void	processDices() Se encarga del lanzamiento de los dados
private void	processMovingPlayer()

	Se encarga del movimiento del jugador por las casillas del tablero
private void	processPlayerBrake() Cuando un jugador cae en una casilla de freno el turno pasará al siguiente jugador.
private void	processPreSplash() Hace de temporizador para mostrar la pantalla de información de la aplicación durante unos segundos
private void	processSplash() Hace de temporizador para mostrar el splash de bienvenida durante unos segundos.
private void	quitAnswerResult() Cuando se contesta a la pregunta correctamente el jugador actual tirará los dados, por el contrario si la pregunta se ha contestado incorrectamente el jugador cambiará de turno sin avanzar casillas
void	run() Método de la interface Runnable.
private void	sleep(int ms) Duerme el hilo principal tantos milisegundos como indicados por parámetro.
private void	startDices() Comienza la animación de lanzar dados
private void	stopDices() Para el lanzamiento de dados y cambia al estado correspondiente
private void	toGame() Se encarga de borrar todos los componentes actuales (botones), de cargar las preguntas, de preparar el tablero, los jugadores y, finalmente, de inicializar el juego
private void	toMainMenu() Se encarga de mostrar el menú principal cargando todos los recursos gráficos que sean necesarios
private void	toOptionsMenu() Se encarga de borrar todos los antiguos componentes del menú anterior y preparar todos los gráficos del menú de opciones.
private void	updateAnswersButtons() Actualiza los botones de las tres posibles respuestas marcando el correspondiente elegido si procede
private void	updateMenuButtons() Actualiza los gráficos de los botones de los módulos
private void	updatePlayersButtons() Actualiza los gráficos de los botones de los jugadores

Question



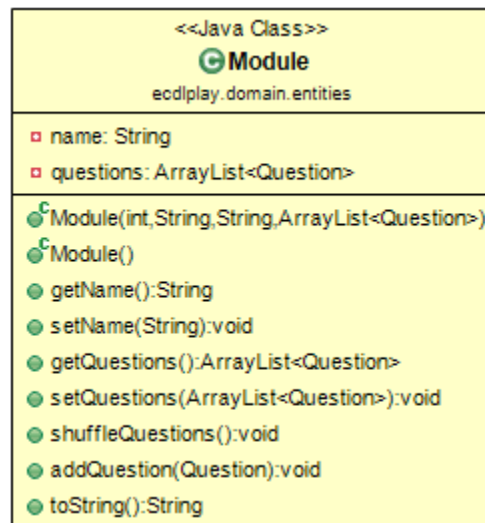
Clase que representa una pregunta en el juego.

Atributos	
private int	dificultad Dificultad de la pregunta
private int	id Identificador de la pregunta
private Image	imagen Imagen asociada a la pregunta
private ArrayList<Answer>	respuestas Lista de posibles respuestas
private String	texto Texto de la pregunta

Constructores
Question() Constructor por defecto de la clase.
Question(int id, String texto, Image imagen, int dificultad, ArrayList<Answer> respuestas) Constructor de la clase con inicialización de las variables

Métodos	
void	addAnswer(Answer answer) Añade una nueva respuesta a la pregunta
int	getDificultad() Obtiene la dificultad de la pregunta
int	getId() Obtiene el identificador de la pregunta
Image	getImagen() Obtiene la imagen asociada a la pregunta
Answer	getRespuesta(int numAnswer) Obtiene la respuesta que corresponde al parámetro
ArrayList<Answer>	getRespuestas() Obtiene el array de respuestas asociadas a la pregunta
String	getTexto() Obtiene el texto de la pregunta
void	setDificultad(int dificultad) Establece la dificultad de la pregunta
void	setId(int id) Establece el identificador de la pregunta
void	setImagen(Image imagen) Establece la imagen asociada a la pregunta
void	setRespuestas(ArrayList<Answer> respuestas) Establece el array de respuestas asociadas a la pregunta
void	setTexto(String texto) Establece el texto de la pregunta
void	shuffleAnswers() Desordena el array de respuestas asociadas a la pregunta

Module



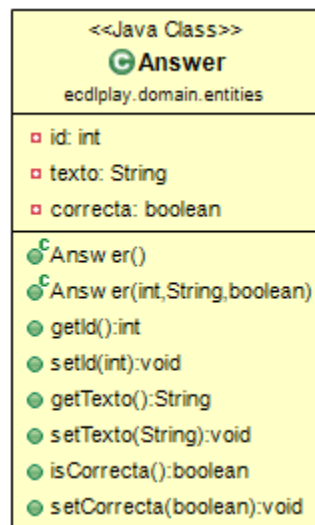
Clase que representa un módulo del juego

Atributos	
private String	name Nombre corto del módulo
private ArrayList<Question>	questions Lista de todas las preguntas asociadas al módulo

Constructores	
Module()	Constructor por defecto de la clase.
Module(int id, String name, String description, ArrayList<Question> questions)	Constructor de la clase con inicialización de las propiedades

Métodos	
void	addQuestion(Question question) Añade una nueva pregunta a la lista
String	getName() Obtiene el nombre corto del módulo
ArrayList<Question>	getQuestions() Obtiene la lista de preguntas del módulo
void	setName(String name) Establece el nombre corto del módulo
void	setQuestions(ArrayList<Question> questions) Establece la lista de preguntas del módulo
void	shuffleQuestions() Mezcla o desordena las preguntas
String	toString() Devuelve una cadena de caracteres concatenando el nombre y la descripción

Answer



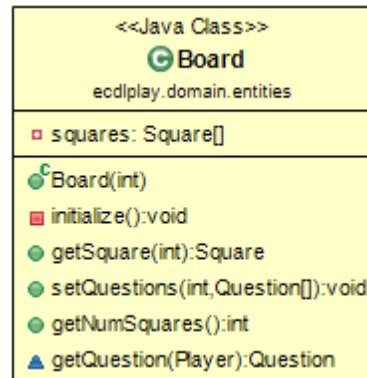
Clase que contiene la información de una respuesta del juego

Atributos	
private boolean	correcta Indica si la respuesta es correcta
private int	id Identificador único de la respuesta
private String	texto Texto de la respuesta

Constructores	
Answer() Constructor por defecto de la clase	
Answer(int id, String texto, boolean correcta) Constructor de la clase con inicialización de variables	

Métodos	
int	getId() Devuelve el identificador de la respuesta
String	getTexto() Devuelve el texto de la respuesta
boolean	isCorrecta() Devuelve si la respuesta es correcta o no
void	setCorrecta(boolean correcta) Establece la corrección de la respuesta
void	setId(int id) Establece el identificador de la respuesta
void	setTexto(String texto) Establece el texto de la respuesta

Board



Clase que representa el tablero del juego.

Atributos

<code>private Square[]</code>	<code>squares</code> Array de casillas dentro del tablero
-------------------------------	--

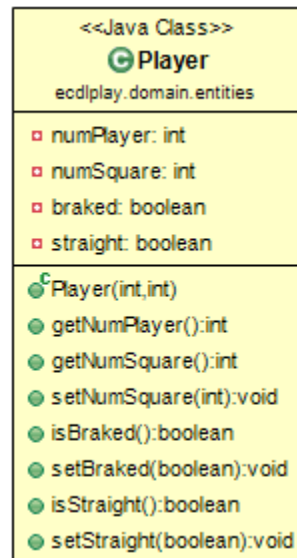
Constructores

`Board(int numSquares)`
Constructor de la clase.

Métodos

<code>int</code>	<code>getNumSquares()</code> Obtiene el número de casillas que formarán el tablero
<code>Question</code>	<code>getQuestion(Player player)</code> Obtiene la pregunta que le corresponde al jugador pasado por parámetro
<code>Square</code>	<code>getSquare(int numSquare)</code> Devuelve la casilla que corresponde al índice pasado por parámetro
<code>private void</code>	<code>initialize()</code> Inicializa cada una de las casillas del tablero
<code>void</code>	<code>setQuestions(int numSquare, Question[] questions)</code> Establece las preguntas que se corresponderán con la casilla pasada por parámetro

Player

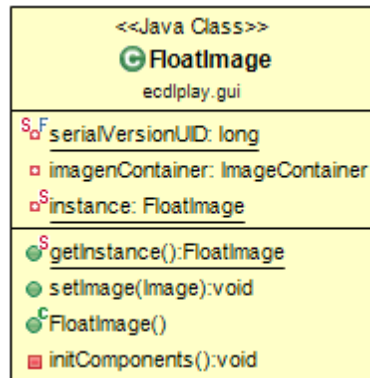


Atributos	
private boolean	braked Indica si el jugador ha caído en una casilla de frenada y no puede jugar el siguiente turno
private int	numPlayer Número de jugador
private int	numSquare Casilla actual dentro del tablero
private boolean	straight Indica si la ficha está en movimiento

Constructores
Player(int numPlayer, int numSquare) Constructor de la clase con inicialización de variables

Métodos	
int	getNumPlayer() Obtiene el número del jugador
int	getNumSquare() Obtiene la casilla en la que se encuentra el jugador
boolean	isBraked() Indica si el jugador está detenido y no puede jugar
boolean	isStraight() Indica si la ficha del jugador está en movimiento
void	setBraked(boolean braked) Establece que el jugador está detenido
void	setNumSquare(int numSquare) Establece la casilla en la que se encuentra el jugador
void	setStraight(boolean straight) Establece que la ficha del jugador está en movimiento

FloatImage



Ventana flotante que servirá para visualizar las imágenes asociadas a preguntas.

Atributos

private ImageContainer	imagenContainer Contenedor de la imagen a mostrar
private static FloatImage	instance Instancia singleton de la ventana

Constructores

FloatImage()	Constructor de la clase. Inicializa la ventana
--------------	--

Métodos

static FloatImage	getInstance() Obtiene o inicializa la instancia singleton
private void	initComponents() Encargada de cambiar el título de la ventana
void	setImage(Image image) Establece la imagen a mostrar

Gestor de preguntas

Pantalla de la aplicación web para gestionar las preguntas de juego:

The screenshot shows a web browser window with the URL `http://localhost:8080/preguntas?mc`. The page title is "ECDL PI@y" and the subtitle is "Editor de Preguntas". There are navigation links for "Home" and "About".

On the left side, under "Opciones", there are links: [Editar módulos](#), [Editar dificultad](#), [Editar Preguntas](#), and [Descargar XML](#).

The main content area is titled "Lista de Preguntas". It features a search form with three fields: "Modulo:" (a dropdown menu), "Dificultad:" (a dropdown menu), and "Texto:" (a text input field). A "Buscar" button is located to the right of these fields.

Below the search form is a table with the following columns: "Acciones", "Texto", "Modulo", and "Dificultad". The table contains 10 rows of questions, all from "Módulo 1" with a difficulty of "Fácil". Each row has an "Acciones" column with a pencil icon and a red minus icon.

Acciones	Texto	Modulo	Dificultad
	¿Cómo se conocen a todas las partes físicas y tangibles de un ordenador?	Módulo 1	Fácil
	Los dispositivos que permiten introducir datos al computador desde el exterior se denominan	Módulo 1	Fácil
	Los periféricos que permiten al computador mostrar datos a los usuarios se denominan	Módulo 1	Fácil
	Los programas informáticos que nos permiten crear documentos como cartas o informes se conocen como	Módulo 1	Fácil
	¿Qué es una LAN?	Módulo 1	Fácil
	¿Qué es Internet?	Módulo 1	Fácil
	El término descarga o download hace referencia a:	Módulo 1	Fácil
	El e-learning consiste en:	Módulo 1	Fácil
	A diferencia del correo electrónico, con la mensajería instantánea:	Módulo 1	Fácil

5 Gestor de preguntas

Realizada en PHP, solo ha sido utilizada para realizar la carga de nuevos módulos, preguntas y respuestas.

Modelo de datos

La base de datos está formada por cuatro tablas:

Ecdl_modulo para almacenar los diferentes modulos, su primary key es un id el cual será foreign key de la tabla ecdl_pregunta.

ID	Identificador único del registro. Clave primaria
NOMBRE	Nombre corto del módulo
DESCRIPCIÓN	Descripción más detallada del módulo
CREATED_AT	Fecha de creación del registro
UPDATED_AT	Fecha de la última modificación del registro

Ecdl_dificultad para almacenar las diferentes dificultades que se quieran poner en el juego, su primary key es un id el cual es foreign_key de la tabla ecdl_pregunta.

ID	Identificador único del registro. Clave primaria
NOMBRE	Descripción de la dificultad
VALOR	Valor numérico de la dificultad.
CREATED_AT	Fecha de creación del registro
UPDATED_AT	Fecha de la última modificación del registro

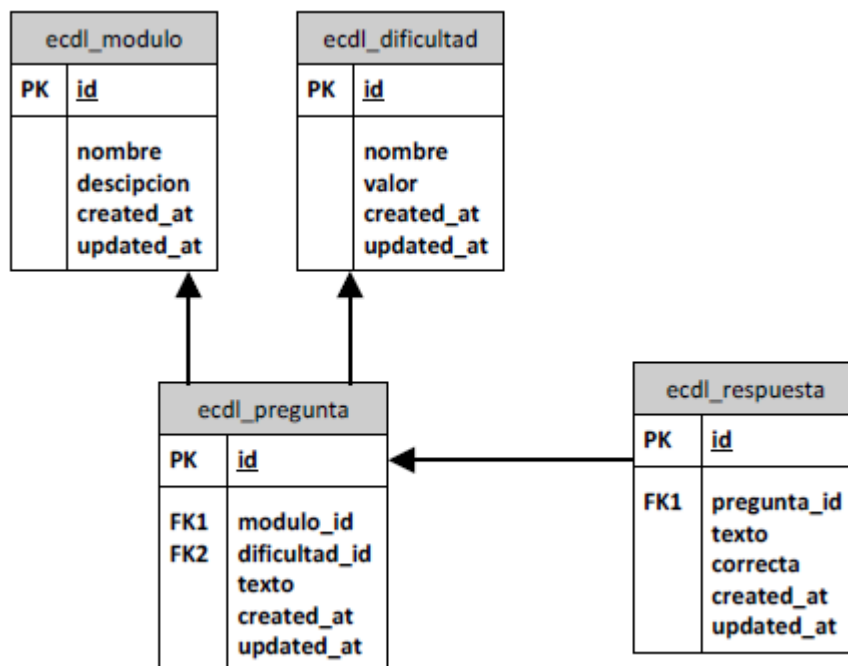
Ecdl_pregunta la cual almacena las diferentes preguntas que tiene el juego, su primary key es un id que es foreign key en la tabla ecdl_respuesta.

ID	Identificador único del registro. Clave primaria
MODULO_ID	Identificador del módulo al que pertenece la pregunta. Clave externa de la tabla ecdl_modulo
DIFICULTAD_ID	Identificador de la dificultad de la pregunta. Clave externa de la tabla ecdl_dificultad.
IMAGEN	Ruta relativa de la imagen asociada a la pregunta. Puede ser nula.

TEXTO	texto de la pregunta
CREATED_AT	Fecha de creación del registro
UPDATED_AT	Fecha de la última modificación del registro

Ecdl_respuesta la cual almacena las diferentes respuestas que hay en el juego.

ID	Identificador único del registro. Clave primaria
PREGUNTA_ID	Identificador de la pregunta a la que corresponde esta respuesta. Clave externa de la tabla ecdl_pregunta
TEXTO	Texto de la respuesta
CORRECTA	Indicador para determinar si la respuesta es correcta o no
CREATED_AT	Fecha de creación del registro
UPDATED_AT	Fecha de la última modificación del registro



Fichero XML

El formato del fichero XML que genera las preguntas sigue el siguiente esquema:

<ecdplay>: Etiqueta de inicio de fichero

<modulo>: Para cada uno de los módulos del ECDL habrá un nodo de este tipo que contendrá los siguientes valores:

<preguntas>: array de nodos de tipo pregunta

<pregunta>: cada una de las preguntas del módulo. Tendrá los siguientes elementos:

<texto>: texto de la pregunta

<dificultad>: dificultad de la pregunta

<imagen>: un texto que representará la imagen asociada a la pregunta en formato Base64. Este nodo puede no existir si la pregunta no tiene imagen

<respuestas>: array de nodos de tipo respuesta

<respuesta>: cada una de las respuestas. La que sea correcta tendrá un atributo cuyo valor será "1"

```
<ecdplay>
  <modulo>
    <id>1</id>
    <nombre>Módulo 1</nombre>
    <descripcion>Conceptos básicos sobre las tecnologías de la información (TI)</descripcion>
    <preguntas>
      <pregunta>
        <texto>El término descarga o download hace referencia a:</texto>
        <dificultad>1</dificultad>
        <respuestas>
          <respuesta correcto="1">La transferencia de archivos desde un emisor hasta nuestro computador.</respuesta>
          <respuesta>La transferencia de archivos desde nuestro ordenador hacia un receptor</respuesta>
          <respuesta>El establecimiento de conexión entre nuestro computador y otro</respuesta>
        </respuestas>
      </pregunta>
      <pregunta>
        <texto>A diferencia del correo electrónico, con la mensajería instantánea:</texto>
        <dificultad>1</dificultad>
        <imagen>/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hy...</imagen>
        <respuestas>
          <respuesta correcto="1">Se produce una comunicación en tiempo real</respuesta>
          <respuesta>Se puede enviar un mensaje a varios destinatarios</respuesta>
          <respuesta>En la mayoría de los casos es posible el envío de archivos</respuesta>
        </respuestas>
      </pregunta>
    </modulo>
    <modulo>...</modulo>
    ...
  </ecdplay>
```

Herramientas y tecnologías utilizadas

Entorno de desarrollo

El entorno de desarrollo utilizado ha sido *Netbeans*, Netbeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.

Existe además un número importante de módulos para extenderlo como por ejemplo el de PHP, que fue utilizado para hacer la aplicación web de gestión de preguntas.



Lenguaje de programación.

El lenguaje de programación para el desarrollo de la aplicación ha sido Java, Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.



Base de datos

El sistema de gestión de base de datos utilizado ha sido *MySQL*. MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.



Máquina virtual

Se ha utilizado una máquina virtual VMWare con *Windows7* donde está alojada la base de datos y la aplicación web de gestor de preguntas.

Las características de la máquina virtual eran 2 GB de Ram, 1 procesador y 60GB de disco.



Control de versiones y repositorio

También se ha contado con un servidor con *Subversion* para tener organizadas las diferentes versiones del juego. Subversion, frecuentemente como SVN, es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD.



Mantenimiento correctivo

Con el fin de detectar los errores que había en la versión anterior de la aplicación, lo que se hizo fue hacer repetidas pruebas. La mejor forma de hacerlas, siendo la aplicación un juego, era precisamente jugar varias partidas.

Lo primero que se hizo fue entrar en la aplicación, nada más entrar se puede ver el menú del juego, en el menú de opciones se puede elegir tanto el módulo que se quiere como el número de jugadores. En las opciones de los módulos no aparecían los nombres de los módulos, lo cual era un poco confuso para el jugador ya que no tendría claro sobre qué tema y preguntas trataría la partida. Por lo tanto, ya había detectado el primer error a solucionar pero antes de ponerme con él quería detectar y observar el resto de errores.

En las opciones de número de jugadores puse la de un jugador y me adentre en el juego.

Una vez dentro del juego fui respondiendo preguntas hasta que salió una con imagen, en la versión anterior sabías si una pregunta tenía imagen porque aparecía un icono con una lupa, al pulsar el icono de la lupa se abría la imagen.

Una vez contestada la pregunta, en la siguiente cuestión aun sin que fuera una pregunta de imagen volvía a aparecer la lupa y pulsando sobre ella volvía a salir la imagen de la pregunta anterior. Este era otro de los errores a corregir.

Continuando el juego y varias partidas con un jugador pase a probar con más jugadores, ahí fue cuando encontré el error más importante de todos.

Jugando con varios jugadores note que había preguntas que se repetían continuamente en un breve periodo de tiempo, algunas incluso seguidas aunque el jugador anterior hubiese contestado correctamente. Este era el peor error de todos pues impedía el buen funcionamiento de la partida así como el fin del juego que es aprender.

El error del menú y los módulos.

Este error fue el más fácil de solucionar, lo más complicado era detectar en que clase y método estaba ubicado el código.

El código estaba en la clase OptionsMenuCanvas en el método paintModules.

Una vez encontrado solo había que cambiar la parte señalada en el código `"module.getName"` y así ya saldrían bien los nombres.

```

for (Module module : GameDataLoader.getLoader().getGameData().getModules()) {

    fontModule.drawString(g, module.getName(),
        GameCanvasConstants.OPTIONS_MENU_MODULE_BOX_X,
        actualY,
        GameCanvasConstants.OPTIONS_MENU_MODULE_BOX_WIDTH,
        GameCanvasConstants.OPTIONS_MENU_MODULE_BOX_HEIGHT,
        TFont.JUSTIFY | TFont.TOP);

    actualY += 40;

}

```

Aquí podemos ver la diferencia entre una versión y otra.



8 ECDL PI@y sin nombre de módulos



9 ECDL PI@y con nombre de modulos

El error de las imágenes y la lupa

Lo que se pretendía solucionando este error era primeramente evitar que después de una pregunta con imagen y saliendo otra pregunta sin imagen apareciera la lupa y la imagen anterior. Por otra parte también se pretendía que en las preguntas con imagen apareciera la imagen sin necesidad de pulsar el icono de la lupa.

Para corregir este error se deshabilitó la función en la cual se mostraba el icono de la lupa y a la hora de obtener una nueva pregunta con sus respuestas se puso una condición en la que si la pregunta contenía una imagen entonces que la mostrara.

```
if(currentQuestion(board.getSquare(players[turn].getNumSquare()).getDifficult()).getI
magen().getImagen() != null)
{ FloatImage.getInstance().setVisible(true);}
```

Esta parte del código aparece en la clase GameEngine en el método getAnswers.

Una vez se pulsa un botón de respuesta la imagen desaparece y no afecta al resto de preguntas venideras.

```
case GameCanvasConstants.BUTTON_ANSWER1:
case GameCanvasConstants.BUTTON_ANSWER2:
case GameCanvasConstants.BUTTON_ANSWER3:
    FloatImage.getInstance().setVisible(false);
```

Esta parte del código aparece en la clase `GameEngine` en el método `processComponentClick`.

Por último, con el fin de que se vieran bien la imagen y el texto de la cuestión, se estableció que la imagen apareciese en un lado de la pantalla y se le puso una ventana variable para poder agrandar o disminuir el tamaño de ella.

```
setResizable(true);
setLocation(1,1);
```

Esta modificación se hizo en la clase `FloatImage` y método de mismo nombre.

Como al corregir el error el icono de la lupa quedaba sin utilidad se optó por retirarlo de la aplicación.

En las siguientes imágenes podemos ver las diferencias entre versiones:



10 ECDL PI@y con icono de lupa



11 ECDL PI@y sin icono de lupa

El error de las preguntas repetidas

Como dije anteriormente, este es el error más importante y más complejo de resolver en este mantenimiento correctivo.

Después de analizar el sistema de preguntas y realizar pruebas me di cuenta de que no era casualidad ni azar el que una pregunta se repitiera en tan poco periodo de tiempo. En realidad no había ningún problema como tal, simplemente el funcionamiento de la aplicación seguía una lógica no adecuada para el tipo de juego.

El funcionamiento y sistema era el siguiente, la aplicación cargaba las preguntas del archivo xml , posteriormente al iniciar el juego lo que hacía era cargar las preguntas del módulo elegido (120 preguntas), barajaba las preguntas y asignaba a cada casilla un numero de preguntas que dependía del número máximo de preguntas que podía tener una casilla, y aquí es donde estaba el problema, al haber 70 casillas y solo 120 preguntas, se le asigna a cada casilla una única pregunta por lo que quedaban sin asignar 50 preguntas.

Por esta razón se repetía preguntas nada más salir, porque ambos jugadores están en la misma casilla, la cual tenía asignada esa pregunta predeterminada, no teniendo otra pregunta alternativa.

Lo que se hizo para solucionar este error fue trasladar el funcionamiento habitual que se haría en un juego de este tipo en la vida real, es decir tendríamos un montón de preguntas, la barajaríamos, cogeríamos la primera pregunta y una vez vista y contestada correctamente la meteríamos al final y cogeríamos la siguiente, si se contestara erróneamente el siguiente jugador tendría la oportunidad de contestar la misma pregunta.

Esto se realiza con listas de preguntas como vemos en el código:

```
private ArrayList<Question> easyQuestions;
private ArrayList<Question> normalQuestions;
private ArrayList<Question> hardQuestions;

private void prepareBoard() {

    // Create Board
    board = new Board(GameEngineConstants.MAX_SQUARES);

    // Shuffle questions
    gd.shuffleQuestions();
    // Get Questions
    easyQuestions=gd.getQuestions(GameEngineConstants.DIFFICULT_EASY,
module);
    normalQuestions=gd.getQuestions(GameEngineConstants.DIFFICULT_NORMAL,
module);
    hardQuestions=gd.getQuestions(GameEngineConstants.DIFFICULT_HARD,
module);

    // Shuffle answers
    for(Question a:easyQuestions){
```

```

        a.shuffleAnswers();
    }
    for(Question a:normalQuestions){
        a.shuffleAnswers();
    }
    for(Question a:hardQuestions){
        a.shuffleAnswers();
    }
}

```

Es un código sencillo en el que en la clase GameEngine se prepara el tablero, aquí era donde antes se asignaba a cada casilla una pregunta, es decir, desde el principio del juego se sabía que pregunta tenía cada casilla.

Ahora lo que se hace es barajar las preguntas, meterlas en sus respectivas listas y barajar las respuestas.

Añadir sonidos al juego

Con el fin de hacer el juego más ameno y divertido se planteó añadir sonidos al juego, tampoco tenía que ser unos sonidos exagerados que llegaran a molestar la jugabilidad, así que se optó por añadir unos cuantos sonidos discretos en las tiradas del dado, en los movimientos de las fichas y en los aciertos y fallos de las respuestas.

Para añadirlos se creó una clase llamada Sound para gestionar el sonido y se llamaban después de cada acción indicada anteriormente.

Mantenimiento Evolutivo

El cambio del syllabus respecto a la versión anterior se basa fundamentalmente en la estructura de los módulos así como en la creación de dos nuevos módulos y la fusión de dos.

A continuación podemos ver las diferencias entre las dos versiones:

El módulo de *Conocimientos básicos de TI* junto con el de *Uso de ordenador* se fusiona en uno creando el nuevo módulo de *Conocimientos Fundamentales de Computación*, además se le añaden nuevas preguntas sobre como conectar un dispositivo y como trabajar con texto.

Se añaden los nuevos módulos de *Colaboración en Línea* y *Seguridad*. A cada nuevo módulo se le añade un total de 120 preguntas repartidas por dificultad teniendo un 30% de ellas imagen en la pregunta.

El módulo de *Información y Comunicación* cambia de nombre pasando a llamarse *Conocimientos Fundamentales de Aplicaciones en Línea*, además del cambio de nombre a este módulo se le añadió preguntas de los temas de uso de calendarios en Outlook.

Los módulos de Excel y Word tienen un pequeño cambio respecto a la versión anterior en algunas referencias.

Nº Anterior	Modulo anterior	Nº Nuevo	Modulo nuevo
1	C. básicos de TI	1	CFC
2	Uso ordenador	3	Word
3	Word	4	Excel
4	Excel	7	CFAL
5	BBDD	5	BBDD
6	PowerPoint	6	PowerPoint
7	Información y comunicación	2	Colaboración en línea
		8	Seguridad

12 Tabla cambios de módulos

Se actualizo el menú de módulos con los dos nuevo módulos y nombres nuevos.

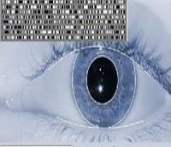
Ejemplo de inserción de una de las nuevas preguntas desde el gestor de preguntas:

Editar Pregunta

Módulo:

Dificultad:

Texto:

Imagen: 

Respuesta 1

Texto:

Correcta: ☐

Respuesta 2

Texto:

Correcta: ☒

Respuesta 3

Texto:

Correcta: ☐

13 Pantalla inserción preguntas

Colaboración en línea – Módulo 2

Objetivos del módulo

Se exige al candidato que entienda los conceptos relacionados con las bases de datos y muestre su competencia en los conceptos claves de las herramientas de colaboración en línea.

El candidato debe ser capaz de:

- Entender los conceptos clave relacionados con la colaboración en línea y computación en nube.
- Configurar cuentas y prepararse para la colaboración en línea.
- Utilizar aplicaciones de almacenamiento en línea y productividad en la web para colaborar.
- Utilizar calendarios en línea y móviles para administrar y planificar actividades.
- Colaborar e interactuar a través de redes sociales, blogs y wikis.
- Programar y dirigir reuniones en línea y utilizar entornos de aprendizaje en línea.
- Entender conceptos clave de tecnología móvil y utilizar funciones de correo electrónico, aplicaciones y sincronización.

Ejemplo de preguntas sobre este módulo

Pregunta de dificultad baja:

¿Cuál de las siguientes características de la colaboración en línea son verdad?

- a) Tiene un acceso en tiempo real a la información**
- b) El alcance es global**
- c) Todas son correctas**

La respuesta correcta es el c), todas son correctas.

El apartado del syllabus del ECDL sobre el que se basa es el siguiente:

Categoría	Área de Conocimiento	Ref.	Unidad de Trabajo
<i>1 Conceptos de colaboración</i>	<i>1.2.Computación en nube</i>	1.2.2	Resumir para los usuarios las ventajas de la computación en nube, por ejemplo: costes reducidos, movilidad ampliada, capacidad de ampliación, actualizaciones automáticas.

Pregunta de dificultad media:

¿Puede modificarse simultáneamente un mismo documento de texto con aplicaciones de productividad en la web?

- a) Sí, es una de las características de dichas aplicaciones**
- b) No, ya que causarían conflictos al escribir dos al mismo tiempo**
- c) No, se crearían varios documentos con las distintas versiones**

La respuesta correcta es la a), es una de las características de dichas aplicaciones.

El apartado del syllabus del ECDL sobre el que se basa es el siguiente:

Categoría	Área de Conocimiento	Ref.	Unidad de Trabajo
<i>3 Utilizar las herramientas de colaboración en línea</i>	<i>3.1 Aplicaciones de almacenamiento en línea y productividad</i>	3.1.5	Identificar las funciones de aplicaciones de productividad basadas en la web: permite la actualización de archivos por usuarios múltiples en tiempo real, permite el uso compartido de archivos.

Pregunta de dificultad alta:

¿Qué velocidad máxima tiene una red 3G?

- a) 5,8 Mbps**
- b) 150 Mbps**
- c) 54 Mbps**

La respuesta correcta es la a), 5,8.

El apartado del syllabus del ECDL sobre el que se basa es el siguiente:

Categoría	Área de Conocimiento	Ref.	Unidad de Trabajo
<i>4 Colaboración móvil</i>	<i>4.1 Conceptos clave</i>	4.1.4	Entender las opciones de conexión de Internet disponibles para dispositivos móviles: inalámbrico (WLAN), Internet móvil (3G, 4G). Entender las opciones asociadas con estas opciones, por ejemplo: velocidad, coste, disponibilidad.

Seguridad - Módulo 8

Objetivos del módulo

Se exige al candidato que muestre su competencia en los conceptos y habilidades esenciales que subyacen al uso seguro de las TIC en la vida diaria y para utilizar técnicas y aplicaciones relevantes que mantengan una conexión de red segura, utilicen Internet de manera segura y gestionen datos e información de manera adecuada.

El candidato debe ser capaz de:

- Comprender los conceptos clave relacionados con la importancia de datos e información seguros, seguridad física, privacidad y robo de identidad.
- Proteger un equipo, dispositivo o red de malware y acceso no autorizado.
- Comprender los tipos de redes, tipos de conexión y problemas específicos de la red, incluidos los firewalls.
- Navegar en la World Wide Web y comunicarse por Internet de manera segura.
- Comprender los problemas de seguridad relacionados con las comunicaciones, incluidos el correo electrónico y la mensajería instantánea.
- Utilizar copias de seguridad y restaurar los datos de manera apropiada y segura, y eliminar datos y dispositivos de manera segura.

Ejemplo de preguntas sobre este módulo

Pregunta de dificultad baja:

¿Cómo se llaman los programas utilizados por los atacantes de un sistema para ocultar puertas traseras que facilitan el acceso y control del sistema infectado?

- a) Troyanos
- b) Rootkits.
- c) Puertas traseras

La respuesta correcta es la b), Rootkits.

El apartado del syllabus del ECDL sobre el que se basa es el siguiente:

Categoría	Área de Conocimiento	Ref.	Unidad de Trabajo
12.2 Malware	12.2.1 Definición y función	12.2.1.2	Reconocer diferentes formas en que puede concebirse el malware, como: troyanos, rootkits y backdoors.

Pregunta de dificultad media:

¿Qué software recopila información de un ordenador y después transmite esta información a una entidad externa sin o con el consentimiento del propietario del ordenador?

- a) Adware
- b) Spyware
- c) Botnets

La respuesta correcta es la b), Spyware.

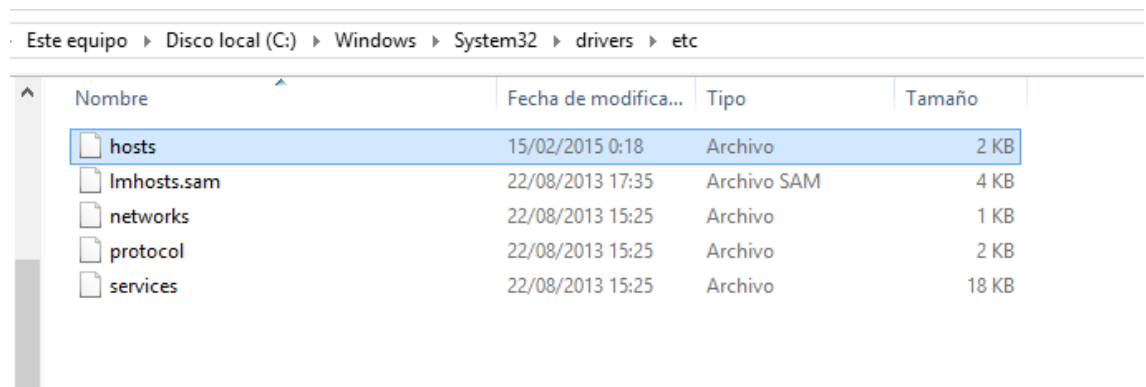
El apartado del syllabus del ECDL sobre el que se basa es el siguiente:

Categoría	Área de Conocimiento	Ref.	Unidad de Trabajo
12.2 Malware	12.2.2 Tipos	12.2.2.2	Reconocer los tipos de robo de datos, malware de extorsión/con ánimo de lucro y comprender cómo trabajan: adware, spyware, botnets, registro de pulsaciones de teclas (keystrokes logging) y marcadores Web (diallers).

Pregunta de dificultad alta:

¿Qué tipo de ataques sufre el archivo señalado en la imagen?

- a) Cracking
- b) Troyanos
- c) Pharming



Este equipo > Disco local (C:) > Windows > System32 > drivers > etc				
Nombre	Fecha de modifica...	Tipo	Tamaño	
hosts	15/02/2015 0:18	Archivo	2 KB	
lmhosts.sam	22/08/2013 17:35	Archivo SAM	4 KB	
networks	22/08/2013 15:25	Archivo	1 KB	
protocol	22/08/2013 15:25	Archivo	2 KB	
services	22/08/2013 15:25	Archivo	18 KB	

La respuesta correcta es la c), Pharming.

El apartado del syllabus del ECDL sobre el que se basa es el siguiente:

Categoría	Área de Conocimiento	Ref.	Unidad de Trabajo
12.4 Uso seguro de la Web	<i>12.4.1 Navegación Web</i>	12.4.1.3	Reconocer el pharming.

Actualización del documento de ayuda

También se ha actualizado el documento de ayuda del juego con las partes modificadas, por ejemplo con la parte de la lupa que ya no sale explicada en el documento.

Pruebas

La siguiente tabla muestra las pruebas realizadas con su resultado pertinente.

Id	Descripción	Tipo	Precondiciones	Acción	Resultado
1	El jugador entra en el menú de módulos y aparecen sus nombres y los nuevos módulos	Interfaz de usuario	Archivo xml en su sitio. Jugador pulsa opciones.	No	Aparecen los módulos con sus nombres y los nuevos módulos
2	El jugador cae en una casillas que tiene una pregunta con imagen, se abrirá la imagen y se mostrara la imagen en un lado de la pantalla	Interfaz de usuario	El jugador ha caído en una casilla pregunta con imagen	No	La ventana de la imagen aparece en un lado de la pantalla
3	El jugador podrá dimensionar la imagen de una pregunta	Interfaz de usuario	El jugador ha caído en una casilla pregunta con imagen	Redimensionar la imagen pinchando sus bordes y deslizando	Ventana de la imagen más grande o pequeña
4	El jugador responde correctamente a una pregunta con imagen e inmediatamente se cierra la ventana de imagen	Interfaz de usuario	El jugador ha caído en una casilla pregunta con imagen	Responder correctamente a la pregunta con imagen	La ventana de la imagen se cierra correctamente
5	El jugador o jugadores caen en una misma casilla y no aparece	Interfaz de usuario	El jugador u otro jugador ha caído ya en la casilla	Caer en la casilla	La pregunta nueva no es repetida

	la misma pregunta				
6	El jugador contesta correctamente a una pregunta y la pregunta no se repite	Interfaz de usuario	El jugador ha caído en una casilla	Responder a la pregunta correctamente	La siguiente pregunta será una diferente
7	El jugador contesta erróneamente a una pregunta y la pregunta se repite	Interfaz de usuario	El jugador ha caído en una casilla	Responder a la pregunta incorrectamente	Se repetirá la pregunta al jugador posterior

14 Tabla pruebas

Conclusiones

El trabajo me ha servido para aprender y darme cuenta de cómo de importante es el mantenimiento.

Como se ha podido comprobar, el mantenimiento es una de las partes más importantes del desarrollo Software, es la parte más infravalorada y que muchas veces se considera poco importante.

Normalmente cuando se piensa en mantenimiento uno solo piensa en el mantenimiento correctivo pero el que más coste y tiempo lleva es el evolutivo. El mantenimiento evolutivo es necesario para tener actualizado el software a las necesidades que produce entidades externas a la aplicación, un software desfasado es poco útil y en un juego de estas características, en la que el tema principal son las nuevas tecnologías y la informática, aún más.

El mantenimiento correctivo también es importante, en la medida que impida el buen funcionamiento de la aplicación, en el caso de esta aplicación el error que más lo impedía era el de la repetición de preguntas, ya no solo por lo molesto de volver a responder una pregunta que acabas de contestar bien, sino porque además impedía que salieran otras preguntas con la que poder aprender.

Para continuar con el mantenimiento de la aplicación, estaría bien en un futuro poder actualizar la aplicación gestora de preguntas y la base de datos, por ejemplo poder acceder a la base de datos directamente desde el juego por medio de internet. Teniendo los juegos como clientes bastaría con actualizar la base de datos para tener actualizados todos los juegos.

En el caso de que fueran muchas maquinas las que tuviesen el juego, esto ahorraría mucho tiempo, también se puede seguir manteniendo este método de carga de preguntas offline en el caso de que no se disponga de internet.

También se podría realizar una aplicación móvil, ya que hoy en día casi todo el mundo dispone de un Smartphone o una Tablet y sería un juego bastante adaptable a estos dispositivos, incluso con la idea de la base de datos online y la conexión a internet, se podrían realizar partidas de varios jugadores online y fomentar dichas partidas con retos o logros virtuales.

En definitiva, esta aplicación tiene muchas posibilidades y puede continuar siendo una gran herramienta para el aprendizaje, además no tiene por qué solo servir para el ECDL ya que la gran portabilidad de las preguntas lo hace perfecto para otros muchos temas de aprendizaje.

Bibliografía

- Diseño y desarrollo de un juego educativo para ECDL, Julio Campuzano

Verjano

- <http://www.ecdl.es/>
- https://es.wikipedia.org/wiki/Mantenimiento_de_software
- http://cnx.org/contents/82e48bb4-de56-44fd-b069-bf470491e5b7@4/Factores_del_coste_del_Ciclo_d
- <http://es.calameo.com/books/003285581c078a5847539>
- https://es.wikipedia.org/wiki/Aprendizaje_electr%C3%B3nico
- <http://www.e-abclearning.com/definicion-learning>
- <http://www.educadictos.com/g-learning-aprender-tambien-jugando/>